Question1

Solution :

First, write N as binary such as 1001….01101, assume it is a list L with K+1 elements (0 or 1), we can get $K \le log_2^n$ and $L_K = 1(\text{ }the\ leftmost\ element)$.

$$So, \qquad N = L_K * 2^K + L_{K-1} * 2^{K-1} + \cdots + L_1 * 2^1 + L_0 * 2^0$$

Then we can compute that $M^1 = M^1 = M^{2^0}$

$$M^1 * M^1 = M^2 = M^{2^1}$$

$$M^2 * M^2 = M^4 = M^{2^2}$$

……..

$$M^{\frac{2}{T}} * M^{\frac{2}{T}} = M^T = M^{2^K}$$

With at most $\lfloor log_2^n \rfloor$ times multiplications we can get $M^{2^0}\ to\ M^{2^K}$.

$$\left(M^{L_K * 2^K}\right) * \left(M^{L_{K-1} * 2^{K-1}}\right) * \ldots * (M^{L_1 * 2^1}) * (M^{L_0 * 2^0})$$

$$= M^{L_K * 2^K + L_{K-1} * 2^{K-1} + \cdots + L_1 * 2^1 + L_0 * 2^0}$$

$$= M^N$$

Proof :

We can get all $M^{2^K}$with $\lfloor log_2^n \rfloor$ times multiplications, then compute $\left(M^{L_K * 2^K}\right) * \left(M^{L_{K-1} * 2^{K-1}}\right) * \ldots * (M^{L_1 * 2^1}) * (M^{L_0 * 2^0})$ with at most $2\lfloor log_2^n \rfloor$ times multiplications. So we can get $M^N$with O(logn) many multiplications.